High-throughput Electron Beam Direct Writing of VIA Layers by Character Projection using Character Sets Based on One-dimensional VIA Arrays with Area-efficient Stencil Design

Rimon IKENO†, Takashi MARUYAMA‡‡, Tetsuya IIZUKA‡‡‡, Satoshi KOMATSU†, Makoto IKEDA†, and Kunihiro ASADA†

†VLSI Design and Education Center (VDEC), The University of Tokyo,
‡‡e-Shuttle, Inc.,
‡‡‡Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo
Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary
Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary
Introduction

Electron-Beam Direct Writing (EBDW)
- **Low-cost** ‘maskless’ exposure solution
  - Mask cost explosion in high-resolution lithography
- **Low exposure throughput**
  - 1 rectangle / 1 EB shot (Variable Shaped Beam; VSB)
- **Limited applications**: low-volume products, test chips, ...

Character Projection (CP) method in EBDW
- Promising approach for **high-throughput** EBDW
- Shooting **multiple figures** at once as a **character**
- **Characters** = “Frequently-appeared layout patterns”
  - Prepared on **character stencil**
- For higher CP throughput → more figures in 1 EB shot
  - **More character varieties required!**
Overview of Character Projection (CP) EBDW

**Shot count comparison**

- **Variable Shaped Beam (VSB)**
  - 1
  - 2
  - 3
  - 4
  - 5

- **Character Projection (CP)**
  - 1
  - 6

**CP EBDW equipment**

- Beam source
- Electron beam
- 1st aperture (beam shaping)
- 2nd aperture (character stencil)
- Characters
- Wafer (resist)
- Exposed character

6x faster throughput \( (E_{CP} = 6.0) \)

CP efficiency: \( E_{CP} \equiv \frac{\text{VSB shot count}}{\text{CP shot count}} \)

Limited stencil area:

→ Limited character variety (number)

Constraint on throughput improvement
Earlier Efforts for Higher CP Throughput

- **Cell layer (transistors & local wires)**
  - 1 cell/shot (major cells only) $\Rightarrow E_{CP} > 10$
  - >1 cells/shot (cell cluster $\rightarrow$ character)
    - cluster extraction / rearrangement

- **Metal layers (interconnect routing)**
  - Tile routing: normalized wire segments
  - Matching to predefined characters

- **VIA layers (interconnect routing)**
  "High degree of freedom"
  - Area-efficient stencil design
    (Du et al./ASP-DAC2012)
    $\Rightarrow$ Just 3 VIAs in 1 character
    $\Rightarrow$ Further improvement?
Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary
VIA Placement Study in Actual Logic Designs

P&R results of example designs (65nm)

<table>
<thead>
<tr>
<th>Design</th>
<th>Area [um²]</th>
<th>Cell#</th>
<th>Cell pin#</th>
<th>VIA1#</th>
<th>VIA1 dens [um⁻²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8080 CPU</td>
<td>11,837</td>
<td>4,293</td>
<td>14,623</td>
<td>14,520</td>
<td>1.23</td>
</tr>
<tr>
<td>USB 2.0</td>
<td>29,309</td>
<td>7,725</td>
<td>28,414</td>
<td>28,301</td>
<td>0.97</td>
</tr>
<tr>
<td>AES Core</td>
<td>60,270</td>
<td>17,328</td>
<td>56,513</td>
<td>56,744</td>
<td>0.94</td>
</tr>
<tr>
<td>DCT Core</td>
<td>63,907</td>
<td>13,800</td>
<td>50,938</td>
<td>50,786</td>
<td>0.79</td>
</tr>
<tr>
<td>SPARC V8</td>
<td>1,191,590</td>
<td>229,580</td>
<td>929,474</td>
<td>927,562</td>
<td>0.78</td>
</tr>
<tr>
<td>SPARC T1</td>
<td>5,933,609</td>
<td>810,278</td>
<td>3,464,043</td>
<td>3,457,402</td>
<td>0.58</td>
</tr>
</tbody>
</table>

- Observations: VIA# and density in actual designs
  - VIA1# ~ cell pin# > cell# → requiring more $E_{CP}$ than cells
  - VIA1 density ~ 1um⁻² → ~1 VIA in 25 grids (1 grid=0.2um)
Problems in Applying CP to VIA Layers

- **Requirements for a VIA CP character set**
  - All possible combinations of multiple VIA arrangement
  - More VIAs in a character for higher CP throughput
  - Integrated within a limited stencil area

- **Reality of the VIA character set planning**
  - ~1 VIA in 25 grids
  - A 4-VIA character occupies 25x4=100 grids in average
  - Character number = Combinatorial number of VIA placement
  - \[ \binom{100}{4} = 3,921,225 \]

- All possible VIA placement: **character number explosion**

- **To achieve higher** \( E_{CP} \) **within stencil area limitation**...
  - Reduce character variations even with more VIAs
  - Detailed VIA placement study (*next page*)
Detailed Study on VIA Arrangement (VIA1)

Relative positions of neighboring VIAs
(USB 2.0, VIA1 layer)

- MAX @ (5,0)
- Ridges @ y = 0, ±9
- Valleys @ y = ±4~5
Detailed Study on VIA Arrangement (VIA2)

Relative positions of neighboring VIAs (USB 2.0, VIA2 layer)

Peaks @ (1, ±3)
A ridge @ y = 0
Gradual decline with increase of x and y
Almost flat ~30,000
Considerations on VIA Arrangement

- **VIA1 layer** (cell pin to interconnect):
  - ridges at y=0, ±9, valleys at y=±4~5
  - The library cell architecture (0.2um x 9 tracks) allows cell pin access (connection) to only 4 tracks in the center
  - VIAs are arranged as stripes to avoid power/ground lines
  - VIA placement has strong correlations in the X direction

- **VIA2 layer**:
  - a ridge at y=0, peaks at (1, ±3)
  - Many cases of the right figures, but also other random cases
Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- **Approach to improve VIA CP throughput**
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary
One Dimensional VIA Array Characters

- Only allow one-dimensional VIA arrays as characters
  - VIA position candidates are reduced
    -> CP character variation decreases

- Two options in VIA array character coding schemes
  - **Binary coding** (fixed character size)
    - “11110”
    - “11101”
    - “11011”
    - “11010”
  - **Span coding** (limited character size & VIA#)
    - “111”
    - “121”
    - “32”
    - “5”

7 shots / 19 VIAs (2.7 VIAs/shot)
Area-efficient Character Stencil Design

Example
Span coding characters
VIA number: N = 4
Max size: M = 6

VIA array characters

Character ordering & superposition

Compressed VIA array

VIA array division & placement

Character stencil

Order code
#1 "111"
#2 "112"
#3 "121"
... 
#19 "141"
#20 "411"

VIA arrangement patterns

"1112122211321312311411"

Example span coding characters
VIA number: N = 4
Max size: M = 6

112122211321312311411

1121222
221132
321312
123114
1411

ASP-DAC 2013 R.Ikeno
Stencil Design under Area & Size Constraints

- Character set planning for stencil design to be adopted to 14nm CP-EBDW equipment specifications
  - Stencil area: $3.6 \times 10^6$ grids / Shot size: 30 grid x 30 grid

Binary coding character
- average character size = 1 grid (after compression)
- character number = $2^{M+1}$ (M: character size)
- $3.6 \times 10^6 < 2^{M+1} \rightarrow M = 20$

Span coding character
- Some options of max VIA number (N) & max character size (M) combination

<table>
<thead>
<tr>
<th>Max VIA# (N)</th>
<th>Max char size (M)</th>
<th>Character number</th>
<th>Stencil area [grid]</th>
<th>Comp. ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>30</td>
<td>621,616</td>
<td>2,894,892</td>
<td>5.80</td>
</tr>
<tr>
<td>8</td>
<td>26</td>
<td>726,206</td>
<td>2,625,930</td>
<td>6.65</td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>880,970</td>
<td>2,689,536</td>
<td>7.43</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>1,097,790</td>
<td>2,989,405</td>
<td>8.13</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
<td>1,401,292</td>
<td>3,319,648</td>
<td>9.19</td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>1,048,576</td>
<td>2,330,190</td>
<td>9.62</td>
</tr>
</tbody>
</table>
Further improvement of CP throughput?

Re-consideration on VIA arrangement:
More VIAs on specific tracks → More VIAs in each shot & less shots for vacant rows

How?

Layout constraints for VIA placement (in detail routing)

VIA1 Layer constraint

VIA NOT allowed
Cell
VIA allowed
VIA NOT allowed

VIA2 Layer constraint

MET1
VIA NOT allowed

MET2
VIA NOT allowed

MET3
Agenda

- Introduction
  - Character Projection (CP) overview
  - Efforts for CP throughput improvement
- Problems in applying CP to VIA layer exposure
  - VIA number, density, and arrangement in actual designs
- Approach to improve VIA CP throughput
  - One-dimensional VIA array characters
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement
- Experimental results
  - CP throughput evaluation
  - CP throughput improvement
- Summary
Experiments and CP Throughput Evaluation

- **Experiments:**
  - Automated place & route of 6 example circuits
  - *with* and *without* the **layout constraints**
    to evaluate impact of introducing the layout constraints

- **CP performance estimation:**
  - **Shot count estimation**
    \[
    (G_{\text{shot/wafer}}) = \frac{10.0 \times 100}{\text{(layout area)}} \times \text{(shot#)}
    \]
  - Average shot area (100Gs/w) \times target shot count = total shot area

- **Evaluation criteria**
  - **Shot count target** for practical use of CP-EBDW @14nm
    \(100\text{~to}173\) Giga shot/wafer
  - 173 Gs/w: the minimum throughput requirement
  - 100 Gs/w: the target to allow margins of other factors
## Results (1): CP Shot Count Estimation

**VIA1 shot# with/wo layout constraint (OpenSPARC T1 Core)**

<table>
<thead>
<tr>
<th>Character Set</th>
<th>N</th>
<th>M</th>
<th>Shot#</th>
<th>VIA# /shot</th>
<th>G shot /wafer</th>
<th>Shot#</th>
<th>VIA# /shot</th>
<th>G shot /wafer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary coding</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td></td>
<td>1,677,663</td>
<td>2.06</td>
<td>282.7</td>
<td>1,032,441</td>
<td>3.32</td>
<td>174.0</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>30</td>
<td>1,301,248</td>
<td>2.66</td>
<td>219.3</td>
<td>858,330</td>
<td>3.99</td>
<td>144.7</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>26</td>
<td>1,437,692</td>
<td>2.40</td>
<td>242.3</td>
<td>916,924</td>
<td>3.74</td>
<td>154.5</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>24</td>
<td>1,570,835</td>
<td>2.20</td>
<td>264.7</td>
<td>986,128</td>
<td>3.48</td>
<td>166.2</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>23</td>
<td>1,597,913</td>
<td>2.16</td>
<td>269.3</td>
<td>994,780</td>
<td>3.44</td>
<td>167.7</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>22</td>
<td>1,627,868</td>
<td>2.12</td>
<td>274.3</td>
<td>1,005,818</td>
<td>3.41</td>
<td>169.5</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>21</td>
<td>1,651,097</td>
<td>2.09</td>
<td>278.3</td>
<td>1,017,748</td>
<td>3.37</td>
<td>171.5</td>
</tr>
<tr>
<td><strong>Span coding</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The layout constraints improved VIA#/shot (=CP efficiency) from **200~300 Gs/w range to 140~175 Gs/w** range
- **Span-coding (N=7, M=30)** achieved the best performance
### Results (2): CP Shot Count Summary

**VIA1/2 shot# of 6 example circuits with layout constraints**

<table>
<thead>
<tr>
<th>Design</th>
<th>VIA1# /shot</th>
<th>VIA1 #shot /wafer</th>
<th>VIA2# /shot</th>
<th>VIA2 #shot /wafer</th>
</tr>
</thead>
<tbody>
<tr>
<td>8080 CPU</td>
<td>5.94</td>
<td>205.7</td>
<td>3.11</td>
<td>311.9</td>
</tr>
<tr>
<td>USB 2.0</td>
<td>6.09</td>
<td>156.2</td>
<td>2.77</td>
<td>272.8</td>
</tr>
<tr>
<td>AES Core</td>
<td>5.64</td>
<td>161.6</td>
<td>2.76</td>
<td>266.7</td>
</tr>
<tr>
<td>DCT Core</td>
<td>4.92</td>
<td>159.2</td>
<td>2.52</td>
<td>248.3</td>
</tr>
<tr>
<td>SPARC V8</td>
<td>4.71</td>
<td>162.9</td>
<td>2.64</td>
<td>269.2</td>
</tr>
<tr>
<td>SPARC T1</td>
<td>3.99</td>
<td>144.7</td>
<td>2.31</td>
<td>236.1</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>4.21</strong></td>
<td><strong>148.1</strong></td>
<td><strong>2.38</strong></td>
<td><strong>242.1</strong></td>
</tr>
</tbody>
</table>

- VIA#/shot: 4~6 (VIA1), 2.3~3.1 (VIA2)
- Shot#/wafer: 148.1 Gs/w (VIA1) $\rightarrow$ Target achieved
  242.1 Gs/w (VIA2) $\rightarrow$ Not achieved

Span-coding: max VIA#: N=7, max size: M = 30
Results (3): Improvement by Tighter Constraints

Further VIA2 throughput improvement?

VIA2 track utility
50% → 33%

VIA2 shot/wafer
270 Gs/w → 208 Gs/w

Wire length increase
2% → 11%

(ASYC V8, span-coding, N=7, M=30)
Introduction
- Character Projection (CP) overview
- Efforts for CP throughput improvement

Problems in applying CP to VIA layer exposure
- VIA number, density, and arrangement in actual designs

Approach to improve VIA CP throughput
- One-dimensional VIA array characters
- Area-efficient stencil design & character set planning
- Layout constraints for further throughput improvement

Experimental results
- CP throughput evaluation
- CP throughput improvement

Summary
Summary

- **Difficulty in applying CP-EBDW to VIA layers**
  - Character explosion when increasing VIA number
  - VIA # / density / arrangement studies in actual designs

- **One-dimensional VIA array characters**
  - Character number suppression by limited arrangement
  - Area-efficient stencil design & character set planning
  - Layout constraints for further throughput improvement

- **Experimental results and conclusions**
  - **Good CP throughput** by span coding character set
  - **Improved CP throughput** with little interconnect impact by tighter layout constraints
  - VIA1: <150 Gs/w → target achieved (100~173 Gs/w)
  - VIA2: ~200 Gs/w → slightly missed / tuning constraints?
Thank you!
Appendices
Required Character Sets for Span Coding

- Characters with size < M are parts of characters with size = M

\[ \text{VIA} \# = n \quad (n < N) \]
\[ \text{size} = m \quad (m < M) \]

\[ \text{VIA} \# = n+1 \quad \text{size} = M \]

Extended to fit to a longer char.

- Required character sets are only following two groups

<table>
<thead>
<tr>
<th>VIA number</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>N-2</th>
<th>N-1</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character size</td>
<td>&lt;M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>o</td>
<td>o</td>
<td>...</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

Group 1: VIA# = N, size ≤ M

Group 2: VIA# < N, size = M

Example: N=4, M=8
Example of Character Ordering Algorithm

Example
Binary coding characters
Max VIA number: N
(= Char size : M + 1)

Character size
M

VIA grid

Start

clist = {all characters}
olist = (0, .. , 0)
order = |clist| (=2^N)
code = “100..0”

olist[code]= order--
h = substr(code, 0, N-1)
i = 1

code = append(i--, h)

Yes

olist[c0]==0?

No

i==0?

End

• Gen character list
• Initialize order storage
• Get the last code and its order (= char#)

• Set character order
• Extract next char head
• Next char tail candidate

• Gen next char candidate

• If not ordered yet, go back to set the order
• If already ordered, try another candidate
## Details of Character Ordering in Page 15

Span coding character set (VIA number: N = 4, Max size: M = 6)

<table>
<thead>
<tr>
<th>No.</th>
<th>Character</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>2</td>
<td>1 1 2 2 1</td>
<td>1 1 2</td>
</tr>
<tr>
<td>3</td>
<td>1 1 3 9</td>
<td>1 1 3</td>
</tr>
<tr>
<td>4</td>
<td>1 1 4 18</td>
<td>1 1 4</td>
</tr>
<tr>
<td>5</td>
<td>1 2 1 3</td>
<td>1 2 1</td>
</tr>
<tr>
<td>6</td>
<td>1 2 2 5</td>
<td>1 2 2</td>
</tr>
<tr>
<td>7</td>
<td>1 2 3 15</td>
<td>1 2 3</td>
</tr>
<tr>
<td>8</td>
<td>1 3 1 13</td>
<td>1 3 1</td>
</tr>
<tr>
<td>9</td>
<td>1 3 2 10</td>
<td>1 3 2</td>
</tr>
<tr>
<td>10</td>
<td>1 4 1 19</td>
<td>1 4 1</td>
</tr>
<tr>
<td>11</td>
<td>2 1 1 8</td>
<td>2 1 1</td>
</tr>
<tr>
<td>12</td>
<td>2 1 2 4</td>
<td>2 1 2</td>
</tr>
<tr>
<td>13</td>
<td>2 1 3 12</td>
<td>2 1 3</td>
</tr>
<tr>
<td>14</td>
<td>2 2 1 7</td>
<td>2 2 1</td>
</tr>
<tr>
<td>15</td>
<td>2 2 2 6</td>
<td>2 2 2</td>
</tr>
<tr>
<td>16</td>
<td>2 3 1 16</td>
<td>2 3 1</td>
</tr>
<tr>
<td>17</td>
<td>3 1 1 17</td>
<td>3 1 1</td>
</tr>
<tr>
<td>18</td>
<td>3 1 2 14</td>
<td>3 1 2</td>
</tr>
<tr>
<td>19</td>
<td>3 2 1 11</td>
<td>3 2 1</td>
</tr>
<tr>
<td>20</td>
<td>4 1 1 20</td>
<td>4 1 1</td>
</tr>
</tbody>
</table>

Compressed VIA array: 1 1 1 2 1 2 2 2 1 1 3 2 1 3 1 2 3 1 1 4 1 1
Results (4): VIA Character Usage Trend vs. Area

Ratio of actually-used characters as a function of design area (cumulative statistics by multiple ICC runs)

VIA character usage is almost proportional to (design area)$^{1/2}$

VIA characters appear randomly

All possible VIA arrangement must be prepared as CP characters

(ASPARC V8, span-coding, N=7, M=30)